

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

Inhalt

1	Einleitung.....	1
1.1	Projektbeschreibung.....	1
1.2	Projektziel.....	1
1.3	Projktbegründung	2
2	Projektplanung.....	2
2.1	Projektphasen	2
2.2	Ressourcenplanung	3
2.3	Entwicklungsprozess.....	3
3	Analysephase	4
3.1	Ist-Analyse	4
3.2	Wirtschaftlichkeit	5
3.3	Projektkosten	5
3.4	Amortisationsrechnung.....	6
3.5	Nicht-monetäre Vorteile.....	7
3.6	Soll-Analyse	7
3.7	Pflichtenheft	7
4	Entwurfsphase	8
4.1	Zielplattform	8
4.2	Architektur	8
4.3	Benutzeroberfläche	9
4.4	Datenbankstruktur	10
4.5	Qualitätssicherung.....	10
5	Implementierungsphase.....	11
5.1	Implementierung Datenbank.....	11
5.2	Implementierung Backend	11
5.3	Implementierung Frontend	12
6	Abnahme und Einführung	12

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

6.1 Abnahme durch den Auftraggeber.....	12
6.2 Deployment.....	13
7 Dokumentation.....	13
8 Fazit.....	14
8.1 Soll-Ist-Vergleich	14
8.2 Lessons Learned.....	14
8.3 Ausblick.....	15
9 Quellenverzeichnis.....	I
10 Glossar	II
11 Anhang	III
11.1 Detaillierte Zeitplanung.....	III
11.2 Verwendete Ressourcen	IV
11.3 Amortisation	IV
11.4 Entity-Relationship-Diagramm	V
11.5 Klassendiagramm.....	V
11.6 Benutzerhandbuch	VI
11.7 Entwicklerdokumentation.....	VII
11.8 Pflichtenheft	VIII
11.9 Auszug C# Quellcode.....	IX

1 Einleitung

Die folgende Projektdokumentation beschreibt den Ablauf eines IHK-Abschlussprojektes, welches im Rahmen einer Ausbildung zum Fachinformatiker Anwendungsentwicklung durchgeführt wurde. Das Projekt wurde in dem Ausbildungsbetrieb COMDOK GmbH durchgeführt, welcher unter anderem ein CRM-Produkt entwickelt und Individualsoftware anbietet. Das CRM-Produkt wird in Form von einer webbasierten Version erneuert und verbessert. Aktuell beschäftigt die COMDOK GmbH 73 Mitarbeiter, wovon 14 in der Softwareentwicklung tätig sind.

Fachbegriffe sind durch Unterstreichung hervorgehoben und können im Glossar (A. 10 Glossar) nachgeschlagen werden.

1.1 Projektbeschreibung

Das durchgeführte Projekt befasst sich mit der Entwicklung einer Web-App zur Sitzplatzbuchung von Pool-Arbeitsplätzen im Unternehmen. Die Web-App soll den Mitarbeitern eine benutzerfreundliche Oberfläche bieten, über die sie auf Basis des internen Active Directory des Unternehmens auf die Anwendung zugreifen. Über ein Dropdown-Menü können Mitarbeiter Etagen auswählen, um die jeweiligen Arbeitsplätze und Sitzplätze in den entsprechenden Büros anzuzeigen. Die Web-App soll eine Buchungsfunktion zur Verfügung stellen, die es Mitarbeitern ermöglicht, einen bestimmten Arbeitsplatz für einen bestimmten Tag zu buchen. Darüber hinaus sollen Mitarbeiter ihre Buchung stornieren können. Die Sicherheit der Anwendung wird durch die Verwendung des internen Active Directory des Unternehmens und geeignete Sicherheitsmaßnahmen gewährleistet. Abschließend wird das Projekt dokumentiert.

Die COMDOK GmbH, vertreten durch den Abteilungsleiter Marc Schmitz, ist der Auftraggeber.

1.2 Projektziel

Das Projekt hat mehrere Ziele, die darauf abzielen, die Nutzung von zukünftigen Pool-Arbeitsplätzen im Unternehmen effizienter zu gestalten. Zunächst soll die Web-App dazu beitragen, überfüllte Büros zu vermeiden, indem sie den Mitarbeitern eine einfache Möglichkeit bietet, verfügbare Arbeitsplätze zu finden und zu buchen. Dadurch können die Büros effizienter genutzt werden und es wird eine bessere Arbeitsumgebung geschaffen. Außerdem soll es die Kosten für das Unternehmen reduzieren und den Mitarbeitern eine flexiblere Arbeitsweise ermöglichen. Durch die Option des Homeoffice können sich die Mitarbeiter den Fahrtweg einsparen und haben die Möglichkeit, von zu

ComBook*Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen*

Hause aus zu arbeiten. Dennoch bleibt eine weitere Option bestehen, und zwar, dass Poolarbeitsplätze geschaffen werden, die von jedem Mitarbeiter genutzt werden können. Das Unternehmen kann dadurch Kosten sparen, indem es beispielsweise weniger Ressourcen für die Anschaffung und Wartung von Arbeitsplätzen benötigt und weniger Stromkosten anfallen.

Ein weiteres Ziel des Projekts ist es, Ordnung und eine zentrale Buchungsübersicht zu schaffen. Die Web-App wird eine klare und übersichtliche Benutzeroberfläche bieten, über die die Mitarbeiter ihre Buchungen verwalten und ändern können. Durch die zentrale Buchungsübersicht wird es einfacher, den Überblick über die verfügbaren Arbeitsplätze zu behalten und doppelte Belegungen eines Arbeitsplatzes zu vermeiden.

Insgesamt wird das Projekt dazu beitragen, die Nutzung von Pool-Arbeitsplätzen im Unternehmen zu optimieren und eine effizientere und kostengünstigere Arbeitsumgebung zu schaffen.

1.3 Projektbegründung

Die Entwicklung einer Web-App für die Sitzplatzbuchung von Pool-Arbeitsplätzen im Unternehmen ist ein wichtiges Projekt, das dazu beitragen wird, die Arbeitsumgebung für die Mitarbeiter zu verbessern, die Effizienz des Unternehmens zu steigern und gleichzeitig Kosten zu sparen. Eine effektive Nutzung von Pool-Arbeitsplätzen bietet zahlreiche Vorteile für Unternehmen, darunter eine bessere Auslastung der vorhandenen Ressourcen, eine Reduzierung der Kosten für das Unternehmen sowie eine flexiblere Arbeitsumgebung für die Mitarbeiter aufgrund von Homeoffice. Die Sitzplatzbuchung von Pool-Arbeitsplätzen bietet hier eine hervorragende Lösung, um den Mitarbeitern mehr Flexibilität zu ermöglichen und gleichzeitig sicherzustellen, dass die vorhandenen Büroflächen effizient genutzt werden.

Darüber hinaus kann die Web-App die Planung und Verwaltung der Arbeitsplätze vereinfachen, welches Zeit und Kommunikation zur Planung spart. Die zentrale Buchungsübersicht erleichtert die Zusammenarbeit zwischen den Mitarbeitern und fördert eine kollaborative Arbeitsumgebung.

2 Projektplanung

2.1 Projektphasen

Es wurden insgesamt 80 Stunden zur Verfügung gestellt, um das Projekt umzusetzen. Diese Stunden wurden vor Projektbeginn auf verschiedene Phasen verteilt, die im

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

Verlauf der Softwareentwicklung durchlaufen werden. Eine Übersicht über die grobe Zeitplanung und die Hauptphasen kann aus Tabelle I entnommen werden.

Projektphase	Geplante Zeit
Analysephase	8h
Entwurfsphase	18h
Implementierungsphase	19h
Testphase	10h
Dokumentation	16h
Abschluss	6h
Zeitreserve	3h
Projektdauer	80h

Tabelle I Zeitplanung

Darüber hinaus wurden die Hauptphasen in kleinere Unterpunkte unterteilt, die in Anhang 9.1: Detaillierte Zeitplanung aufgelistet sind.

2.2 Ressourcenplanung

Im Anhang (A. 9.2: Verwendete Ressourcen) befindet sich eine Übersicht aller Ressourcen, die für die Durchführung des Projekts eingesetzt wurden. Dazu gehören sowohl Hard- und Software-Ressourcen als auch das eingesetzte Personal. Bei der Auswahl der verwendeten Software wurde darauf geachtet, dass sie entweder kostenfrei verfügbar ist (z.B. Open Source) oder dass die notwendigen Lizenzen schon verfügbar waren. Auf diese Weise konnten die Projektkosten so niedrig wie möglich gehalten werden.

2.3 Entwicklungsprozess

Das Projekt zur Entwicklung der Web-App für die Sitzplatzbuchung von Pool-Arbeitsplätzen im Unternehmen folgte dem erweiterten Wasserfallmodell als Entwicklungsprozess. Die Vorteile des erweiterten Wasserfallmodells sind, dass die Phasen strikt voneinander getrennt werden. Außerdem wird man etwas flexibler, da man, im Gegensatz zum einfachen Wasserfallmodell, in den Phasen vor und zurückspringen kann.

In der ersten Phase des Projekts wurden die Anforderungen und Ziele des Projekts gemäß dem Pflichtenheft (A. 9.8 Pflichtenheft) festgelegt, um eine klare Vorstellung davon zu erhalten, was die Web-App leisten soll und wie sie aussehen sollte. Dabei wurden auch die technischen Anforderungen und Einschränkungen berücksichtigt.

ComBook*Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen*

In der nächsten Phase wurde ein Entwurfskonzept für die Web-App erstellt, welches alle wichtigen Funktionen und Aspekte der Anwendung umfasste. Dabei wurden auch die Benutzeroberfläche und das Design berücksichtigt. Ebenfalls wurde die Datenbankstruktur entwickelt. Es wurden alle notwendigen Datenfelder und Beziehungen zwischen den Tabellen festgelegt, um sicher zu stellen, dass alle Daten effektiv und sicher gespeichert werden können.

Anschließend wurde mit der Implementierung der Web-App begonnen, wobei die verschiedenen Module und Funktionen der Anwendung entwickelt und getestet wurden. Dabei wurde auch die Integration mit dem Active Directory berücksichtigt, um sicherzustellen, dass die Anwendung sicher und nahtlos verwendet werden kann.

Nach der Implementierung folgte eine Testphase, in der sowohl Funktionstests als auch Black- und Whitebox Tests und Benutzertests durchgeführt wurden, um sicherzustellen, dass die Anwendung reibungslos funktioniert und alle Anforderungen erfüllt werden.

Zum Abschluss des Projekts wurde die Web-App vom Auftraggeber abgenommen und danach in Betrieb genommen. Ein Wartungsplan wurde erstellt, um sicherzustellen, dass die Anwendung auch in Zukunft reibungslos funktioniert und bei Bedarf aktualisiert werden kann.

3 Analysephase

3.1 Ist-Analyse

Die COMDOK GmbH verfügt derzeit über ein Arbeitsgebäude, das für etwa 60 Mitarbeiter ausgelegt ist. Während der COVID-19-Pandemie sind viele neue Mitarbeiter dazugekommen und haben die Option des Homeoffice genutzt, was zu keinem Arbeitsplatzmangel führte. Allerdings ist die Homeoffice-Regelung der Bundesregierung inzwischen ausgelaufen und nun hat nicht jeder Mitarbeiter einen festen Sitzplatz, da es nicht genügend Arbeitsplätze gibt. Die Einführung von Homeoffice-Regelungen für zwei Tage pro Woche hat das Problem nicht gelöst, da es keine zentrale Übersicht über die Belegung der Büroarbeitsplätze gibt. Infolgedessen kann es an einigen Tagen zu überfüllten Büros kommen.

Derzeit gibt es keine zentrale Buchungsübersicht für die Büroarbeitsplätze, was die Verwaltung und Organisation der Arbeitsplätze erschwert. Es ist daher notwendig, eine Web-App zur Sitzplatzbuchung zu entwickeln, die es den Mitarbeitern ermöglicht, ihre Arbeitsplätze zu reservieren und die Verwaltung der Arbeitsplätze zu vereinfachen.

3.2 Wirtschaftlichkeit

Das Projekt zur Entwicklung einer Web-App für die Sitzplatzbuchung von Pool-Arbeitsplätzen im Unternehmen ist mit einem gewissen Aufwand verbunden. Es müssen Kosten für die Entwicklung und Implementierung der Software anfallen. Demgegenüber wird das Projekt voraussichtlich auch einige Kosteneinsparungen für das Unternehmen bringen.

Durch den Einsatz von Pool-Arbeitsplätzen kann das Unternehmen die Arbeitsplätze effizienter nutzen. Darüber hinaus können auch die Kosten für die Einrichtung und Wartung von Arbeitsplätzen reduziert werden, da die Mitarbeiter nur noch auf allgemein zugängliche Arbeitsplätze zugreifen können, anstatt individuelle Arbeitsplätze mit eigenen Geräten und Zubehör auszustatten. Ebenfalls wird durch Pool-Arbeitsplätze auch weniger Strom verbraucht, da allgemein weniger Arbeitsplätze vorhanden sind und diese nicht jeden Tag belegt werden. Mitarbeiter sparen sich ebenfalls Kosten und Zeit, da Homeoffice angeboten wird, wodurch der Arbeitsweg wegfällt.

3.3 Projektkosten

Die Projektkosten, die während der Entwicklung des Projektes anfallen, sollen im Folgenden kalkuliert werden. Dafür müssen neben den Personalkosten, die durch die Realisierung des Projektes verursacht werden, auch noch Stromkosten berücksichtigt werden. Da die genauen Personalkosten nicht veröffentlicht werden dürfen, werden kalkulatorische Stundensätze genutzt. In den Stundensätzen sind das Bruttogehalt, der Arbeitgeberanteil der Sozialversicherung sowie die Kosten für den Arbeitsplatz und Energiekosten berücksichtigt. Die Kosten, die für die einzelnen Vorgänge des Projektes anfallen, sowie die gesamten Projektkosten lassen sich der Tabelle II: Projektkosten entnehmen.

Kostenpunkt	Mitarbeiter	Stundensatz	Stunden	Kosten
Entwicklungskosten	1x Auszubildender	50,00€	80	4.000,00€
Blackbox-Test	2x Mitarbeiter	100,00€	2	400,00€
Abnahme	1x Abteilungsleiter	200,00€	2	400,00€
Projektkosten gesamt:				4.800,00€

Tabelle II - Projektkosten

3.4 Amortisationsrechnung

Im nächsten Abschnitt wird die Wirtschaftlichkeit des Projekts untersucht, um zu bestimmen, ab welchem Zeitpunkt sich die Investitionskosten für die Entwicklung der Anwendung amortisiert haben. Die Amortisationsdauer gibt an, wie lange es dauert, bis die Einsparungen durch das neue Produkt die Anschaffungskosten wieder ausgleichen. Dies ermöglicht eine Beurteilung der Wirtschaftlichkeit des Projekts und zeigt, ob es langfristige Kostenvorteile bietet.

Durch die Implementierung einer Buchungswebsite für Pool-Arbeitsplätze kann das Unternehmen den Arbeitsplatzbedarf effektiver verwalten und die Anzahl der fest zugewiesenen Arbeitsplätze reduzieren. Dadurch können Kosten für Anschaffung, Wartung und Stromverbrauch gespart werden. Ebenfalls soll hiermit Zeit gespart werden, die ein Mitarbeiter verbraucht, um herauszufinden, welcher Sitzplatz noch nicht belegt ist.

Um die folgende Amortisationsrechnung zu vereinfachen, wird nur die Zeit der Mitarbeiter betrachtet. Im Schnitt verbringen täglich 10 Mitarbeiter jeweils knapp drei Minuten, um herauszufinden, ob ein Arbeitsplatz belegt ist und welcher frei ist. Das bedeutet eine Zeitersparnis von 30 Minuten am Tag. Wenn man mit einem Stundensatz von 100 € pro Stunde rechnet, dann ergibt es eine Kostenersparung von 50 € pro Tag ($100 \text{ €/h} \cdot 0,5 \text{ h} = 50 \text{ €/Tag}$). Bei einem Durchschnitt von 21 Arbeitstagen pro Monat wäre es eine Einsparung von 1.050 € pro Monat.

Berechnung der Amortisationsdauer:

$$30 \frac{\text{min}}{\text{Tag}} \times 21 \frac{\text{Tag}}{\text{Monat}} = 630 \frac{\text{min}}{\text{Monat}} = 10,5 \frac{\text{h}}{\text{Monat}}$$

$$10,5 \frac{\text{h}}{\text{Monat}} \times 100 \text{ €} = 1050 \frac{\text{€}}{\text{Monat}}$$

$$\frac{4800 \text{ €}}{1050 \text{ €}} = 4,57 \text{ Monate} \approx 139 \text{ Tage}$$

Außerdem wurde die Amortisation graphisch dargestellt (A. 9.3 Amortisation). So soll die Amortisationsdauer noch einmal veranschaulicht werden.

Durch die Amortisationsrechnung lässt sich schließen, dass das Projekt nach 139 Tagen die Kosten ausgeglichen hat. Da die Umstellung auf Pool-Arbeitsplätze eine langfristige Änderung ist, kann man das Projekt auch unter wirtschaftlichen Gesichtspunkten als sinnvoll einstufen.

ComBook*Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen*

3.5 Nicht-monetäre Vorteile

Die Sitzplatzbuchung bietet den Mitarbeitern die Flexibilität, je nach Bedarf verschiedene Arbeitsplätze zu nutzen und vom Homeoffice aus zu arbeiten. Dadurch können Mitarbeiter Zeit und Geld sparen und haben eine bessere Work-Life-Balance. Durch die Möglichkeit, von zu Hause aus zu arbeiten und die Anzahl der notwendigen Fahrten ins Büro zu reduzieren, kann das Projekt zu einer Verringerung des CO₂-Ausstoßes beitragen. Dies wiederum kann einen positiven Effekt auf die Nachhaltigkeit des Unternehmens haben und auch einen Beitrag zum Umweltschutz leisten. Außerdem kann die höhere Flexibilität zu einer höheren Mitarbeiterzufriedenheit und Produktivität führen.

3.6 Soll-Analyse

Im Rahmen der Lösung des beschriebenen Problems soll eine webbasierte Lösung eingeführt werden, die es den Mitarbeitern ermöglicht, ihre Sitzplätze selbstständig zu buchen. Durch diese Lösung wird es möglich sein, die Bürobelegung effektiver zu planen und zu kontrollieren, insbesondere da das Unternehmen plant, von individuellen Arbeitsplätzen auf Pool-Arbeitsplätze umzustellen und vermehrt auf Homeoffice zu setzen. Die interne Web-App wird eine benutzerfreundliche Oberfläche bieten, über die die Mitarbeiter ihre Sitzplätze schnell und einfach buchen, einsehen und stornieren können. Die Anmeldung erfolgt über das intern bestehende Active Directory des Unternehmens, das die Schnittstelle zur Web-App bereitstellt. Die Buchungen werden in einer Datenbank gespeichert und können jederzeit abgerufen werden. Die Etagen des Gebäudes werden in der Web-App visuell dargestellt, so dass die Mitarbeiter schnell freie Sitzplätze finden können. Darüber hinaus wird jedes Büro angezeigt, sodass eine genaue Übersicht über die verfügbaren Sitzplätze geboten wird. Durch die Einführung dieser Lösung wird es möglich sein, Überbelegungen zu vermeiden und die Bürobelegung effektiver zu steuern.

3.7 Pflichtenheft

Zum Schluss wurde ein Pflichtenheft erstellt, in denen die Anforderungen beschrieben sind und umgesetzt werden. Es dient als Leitfaden für die Entwicklung des Projekts. Ein Auszug aus dem Pflichtenheft ist im Anhang (A. 9.8 Pflichtenheft) zu finden.

4 Entwurfsphase

4.1 Zielplattform

In der Entwurfsphase des Projekts wurde als Zielplattform eine webbasierte Anwendung definiert, die auf React als Frontend-Technologie, C# als Backend-Technologie und MSSQL als Datenbank aufbaut. Die Kommunikation zwischen Frontend und Backend erfolgt über eine REST API, die auf dem MVC-Pattern basiert. Dabei werden die Daten zwischen Frontend und Backend im JSON-Format ausgetauscht. Die Wahl dieser Technologien erfolgte aufgrund ihrer weit verbreiteten Verwendung, um eine hohe Kompatibilität und Flexibilität zu gewährleisten und um eine Skalierbarkeit und Erweiterbarkeit der Anwendung sicherzustellen. Durch die Nutzung von React als Frontend-Technologie kann eine hohe Benutzerfreundlichkeit und ein responsives Design gewährleistet werden, während C# als Backend-Technologie eine sichere und effiziente Datenverarbeitung ermöglicht. Die Verwendung von MSSQL als Datenbank ermöglicht eine robuste und zuverlässige Datenhaltung. Die Verwendung von MSSQL gegenüber einer anderen Datenbanklösung ist damit zu begründen, dass das Unternehmen schon MSSQL als Datenbank benutzt und somit keine Mehrkosten entstehen würden und die Erstellung eines Datenbankservers somit wegfällt. Die Implementierung einer REST API, basierend auf dem MVC-Pattern, ermöglicht eine klare Trennung von Daten, Präsentation und Logik, was zu einer hohen Wartbarkeit und Erweiterbarkeit der Anwendung beiträgt.

4.2 Architektur

Die Architektur des Projekts basiert auf einer Client-Server-Architektur, wobei der Client eine webbasierte Benutzeroberfläche mit React als Frontend-Technologie nutzt und der Server in C# implementiert ist. Die Kommunikation zwischen Client und Server erfolgt über eine REST API-Schnittstelle.

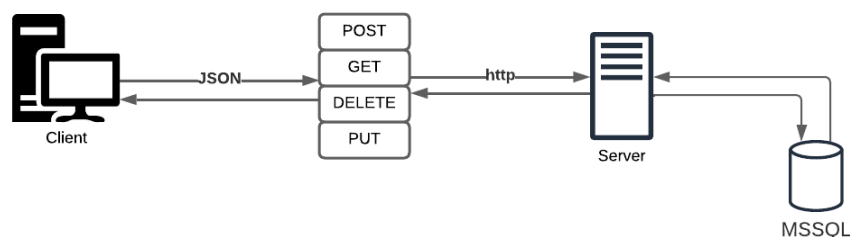


Abbildung I – Skizze-Serverarchitektur

ComBook*Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen*

Im Backend wurden verschiedene Schichten implementiert, um eine klare Trennung von Verantwortlichkeiten und eine hohe Flexibilität zu gewährleisten. Die Schichten umfassen die Modelschicht, Datenzugriffsschicht und die Logikschicht.

Die Logikschicht enthält die Kernfunktionalität des Systems. Hier wird die Logik implementiert, die die Buchungen definiert und die Datenvalidierung durchführt. Die Schicht ist so strukturiert, dass sie durch Austauschbarkeit der einzelnen Komponenten einfach erweiterbar und testbar ist.

In der Präsentationsschicht wird das User-Interface des Systems implementiert. Hier wird React als Frontend eingesetzt, um eine dynamische und benutzerfreundliche Benutzeroberfläche bereitzustellen. Die Schicht ist so strukturiert, dass die UI-Komponenten einfach wiederverwendet und skaliert werden können.

Zur Umsetzung dieser Architektur wurde das MVC-Pattern angewandt. Dabei werden die einzelnen Schichten klar getrennt, um die Verantwortlichkeiten der jeweiligen Komponenten zu definieren und eine klare Strukturierung des Codes zu gewährleisten. Außerdem wird ein hohes Maß an Flexibilität und Skalierbarkeit erreicht, sodass das System leicht erweitert und angepasst werden kann, ohne die Kernfunktionalitäten zu verändern.

4.3 Benutzeroberfläche

Die Benutzeroberfläche des Projekts wurde unter Berücksichtigung der Benutzerfreundlichkeit und des Bedienkomforts entwickelt. Ein Kalender wurde als primäres Werkzeug für die Sitzplatzbuchung gewählt, um die Auswahl der gewünschten Tage für die Benutzer so einfach wie möglich zu gestalten.

In der Oberfläche werden die bereits belegten Sitzplätze in roter Farbe angezeigt, während die eigenen Buchungen in orangener Farbe hervorgehoben werden. Freie Sitzplätze werden in Weiß und ausgewählte in Grün angezeigt. Auf diese Weise können Benutzer schnell einen Überblick über die Verfügbarkeit der Sitzplätze für den gewünschten Zeitraum erhalten und ihre Buchungen entsprechend planen.

Die Oberfläche ist so gestaltet, dass sie intuitiv zu bedienen ist und Benutzer in der Lage sind, schnell und einfach Sitzplätze zu buchen, zu ändern oder zu stornieren. Die Verwendung von Farben trägt dazu bei, dass der Buchungsprozess visuell ansprechender und leichter verständlich wird.

Darüber hinaus wird die Benutzeroberfläche automatisch aktualisiert, um sicherzustellen, dass Benutzer stets die aktuellsten Informationen zu den verfügbaren

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

Sitzplätzen erhalten. Dies reduziert die Möglichkeit von Doppelbuchungen und sorgt für einen reibungslosen Buchungsprozess.

4.4 Datenbankstruktur

Im Rahmen des Projekts wurde eine Datenbank erstellt, um die Buchungen der Arbeitsplätze und Räume zu speichern. Die Datenbank besteht aus drei Haupttabellen: "booking", "room" und "workplace".

Die Tabelle "booking" enthält Informationen über jede Buchung, einschließlich des Zeitraums der Buchung, des Benutzers, der den Arbeitsplatz reserviert hat, das Büro und des Arbeitsplatzes selbst. Diese Tabelle ist die zentrale Tabelle des Systems und dient dazu, die Belegung der Arbeitsplätze und Räume zu verwalten und um doppelte Buchungen zu vermeiden.

Die Tabelle "room" enthält Informationen über jeden Raum im Gebäude. Dazu gehört der Name des Raums und die Etage, auf der man sich befindet. Diese Tabelle wird verwendet, um sicherzustellen, dass die Buchungen dem richtigen Raum und der richtigen Etage zugeordnet sind.

Die Tabelle "workplace" enthält Informationen über jeden Arbeitsplatz im Gebäude. Dazu gehört die Nummer des Arbeitsplatzes und der Raum, in dem sich der Arbeitsplatz befindet.

Das Entity-Relationship-Diagramm (ERD) für die Datenbankstruktur ist im Anhang (A. 9.4 ERD) der Dokumentation zu finden und zeigt die Beziehungen zwischen den Tabellen und den einzelnen Feldern. Durch die Verwendung dieses ERD kann die Struktur der Datenbank leicht verstanden und verwaltet werden.

4.5 Qualitätssicherung

Die Sicherstellung der Qualität während der Entwicklung ist ein wichtiger Bestandteil des Projekterfolgs. Um sicherzustellen, dass das Projekt qualitativ hochwertig umgesetzt wird, wurden verschiedene Maßnahmen ergriffen.

Zu diesen Maßnahmen gehörten insbesondere Black- und White-Box-Tests, die dazu dienten, die korrekte Funktionsweise der Anwendung sicherzustellen. Während der Black-Box-Tests wurden dabei die funktionalen Anforderungen der Anwendung überprüft, um sicherzustellen, dass diese korrekt umgesetzt wurden. Bei den White-Box-Tests hingegen wurden die technischen Aspekte der Anwendung wie z.B. die Programmlogik und der Code getestet. Darüber hinaus wurden auch während der

Programmierung manuelle Tests durchgeführt, um sicherzustellen, dass die Anwendung stabil und benutzerfreundlich ist. Dabei wurden insbesondere die Benutzerfreundlichkeit und die intuitiv verständliche Bedienung der Anwendung getestet, um sicherzustellen, dass die Anwender eine positive Erfahrung bei der Nutzung der Anwendung haben.

5 Implementierungsphase

5.1 Implementierung Datenbank

Basierend auf dem Datenmodell, das in Abschnitt 4.4 (Datenbankstruktur) für die zu erstellende Anwendung definiert wurde, wurde die Implementierung der Datenbank durchgeführt. Zunächst wurde ein neues MSSQL-Datenbank-Schema mit dem Namen "combook" erstellt. Anschließend wurden die benötigten Tabellen mithilfe von SQL-Statements angelegt. Hierfür wurde das Tool Microsoft SQL Server verwendet. Als Grundlage für die Struktur diente das in der Entwurfsphase erstellte ERD, welches im Anhang (A. 9.4: ERD) abgebildet ist. Die erstellte Datenbank entspricht somit der in der Entwurfsphase definierten Struktur und erfüllt die nötigen Anforderungen.

5.2 Implementierung Backend

Für die Implementierung des Backends wurde zunächst auf Basis des in der Entwurfsphase erstellten Klassendiagramms (A. 9.5: Klassendiagramm) das Modell der Anwendung umgesetzt.

Zur Umsetzung des Backends wurden die Programmiersprache C# und das .NET Framework genutzt. Die REST-API wurde mittels ASP.NET Core umgesetzt. Hierbei wurden auch die für die Anwendung notwendigen Packages installiert und konfiguriert.

Es wurden drei verschiedene Models (Room, Workplace, Booking) mit jeweils drei verschiedenen Controllern (RoomController, WorkplaceController, BookingController) erstellt, die für die Büroerstellung, Arbeitsplatzerstellung und Buchung nötig sind. Die Hauptlogik ist in der Klasse ConnectDataBase vorhanden. Diese regelt sowohl die Verbindung zur Datenbank als auch die Verarbeitung von den Daten. In der Methode GetRooms(Room) werden die Räume aus der Datenbank zu einer bestimmten Etage geholt. Die SQL-Abfragen sind im Code definiert, da die Komplexität der Abfragen für eine Stored Procedure zu gering sind und so das Debugging einfacher möglich ist. Das Model Room erhält eine Liste von Workplace, welche eine roomList in Form von JSON an das Frontend zurückgibt. Die Methode BookWorkPlace(Booking) schreibt einen Eintrag in die Datenbanktabelle booking. Die Methode

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

`DeleteWorkPlace(Booking)` storniert die ausgewählte Buchung in Form einer DELETE Anweisung. Einen Auszug aus der Klasse `ConnectDataBase` befindet sich im Anhang (A. 11.9 Auszug C# Quellcode).

5.3 Implementierung Frontend

Die Implementierung des Frontends wurde mit Hilfe des React-Frameworks durchgeführt. Das Frontend fungiert als View im MVC-Pattern und kommuniziert mit dem Backend über eine REST API.

Zunächst wurde eine Login-Funktion implementiert, die es den Benutzern ermöglicht, sich sicher in die Anwendung einzuloggen. Die Login-Daten werden über die REST-API Schnittstelle an das Backend gesendet und gegen das Active Directory geprüft. Bei erfolgreicher Authentifizierung wird eine Authentifizierungs-Session gestartet.

Anschließend wurde ein Kalender implementiert, der dem Benutzer ermöglicht, die Tage schnell und intuitiv auszuwählen. Der Kalender ist benutzerfreundlich gestaltet und ermöglicht eine einfache Navigation zwischen den Tagen, Monaten und Jahren. Die Arbeitsplätze und die Büroräume werden aus der Datenbank genommen, um diese schnell anzulegen und da es eine einfache Möglichkeit bietet, diese zu bearbeiten. Die Büros werden auf der Oberfläche automatisch positioniert und die Arbeitsplätze werden den jeweiligen Büros zugeordnet.

Die ausgewählten Daten werden über die REST API an das Backend gesendet, wo sie verarbeitet und in der Datenbank gespeichert werden. Das Frontend zeigt auch die bereits gebuchten Sitzplätze anhand einer Farbcodierung an, um dem Benutzer einen Überblick über die Verfügbarkeit zu geben.

Das Frontend wurde unter Berücksichtigung von Webdesign-Prinzipien gestaltet, um eine optimale Benutzererfahrung zu gewährleisten. Die Implementierung wurde auch unter Berücksichtigung der Kompatibilität mit verschiedenen Browsern und Endgeräten durchgeführt.

6 Abnahme und Einführung

6.1 Abnahme durch den Auftraggeber

Nachdem die Anwendung fertig gestellt war, wurde Sie dem Abteilungsleiter zur Endabnahme vorgelegt. Bei der Endabnahme wurden verschiedene Aspekte der Anwendung überprüft. Hierzu zählte zum Beispiel die Einhaltung der im Pflichtenheft definierten Anforderungen und Ziele, die Funktionalität der Anwendung sowie deren

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

Benutzerfreundlichkeit und Stabilität. Es wurde geprüft, ob alle Funktionen und Module der Anwendung einwandfrei funktionieren und ob die Anwendung in der Lage ist, den Benutzern eine intuitive und effiziente Nutzung zu ermöglichen. Zudem wurde überprüft, ob die Anwendung sicher und robust genug ist, um den Anforderungen des Unternehmens gerecht zu werden. Durch die Endabnahme konnte sichergestellt werden, dass die Anwendung den Anforderungen entspricht und den Mitarbeitern ein gutes Werkzeug für die Buchung von Arbeitsplätzen bietet.

6.2 Deployment

Um die neue Sitzplatzbuchungs-Web-App für jeden Mitarbeiter erreichbar zu machen, wird die Anwendung intern gehostet. Es wurde eine E-Mail an jeden Mitarbeiter verschickt, die das Benutzerhandbuch und den Link beinhaltet. So konnte sich jeder Mitarbeiter an der Web-App mit dem bekannten Nutzernamen und Passwort aus dem Active Directory anmelden und seine Sitzplätze buchen. Von einer Benutzerschulung wurde abgesehen, da das Benutzerhandbuch jede Funktion erklärt und die Anwendung intuitiv gestaltet wurde.

7 Dokumentation

Im Benutzerhandbuch (A.9.6: Benutzerhandbuch) wird die Anwendung aus der Perspektive des Endnutzers beschrieben. Hier werden die verschiedenen Funktionen und deren Nutzung detailliert erläutert. Hierzu gehören beispielsweise Anweisungen zum Login, zur Auswahl der Etagen und zur Buchung eines Sitzplatzes. Außerdem enthält das Benutzerhandbuch Informationen zu den Systemanforderungen sowie Hinweise zur Fehlerbehebung und zur Kontaktaufnahme mit dem Support-Team.

Die Entwicklerdokumentation (A.9.7: Entwicklerdokumentation) richtet sich an Entwickler, die an der Weiterentwicklung oder Wartung der Anwendung beteiligt sind. Hier werden die technischen Details der Implementierung beschrieben. Dazu gehören beispielsweise die verwendeten Technologien und Frameworks, die Architektur der Anwendung, die Datenbankstruktur sowie die API-Schnittstellen und ihre Funktionen. Außerdem enthält die Entwicklerdokumentation Hinweise zur Code-Struktur und sowie zur Konfiguration und zum Deployment der Anwendung.

8 Fazit

8.1 Soll-Ist-Vergleich

Nach Abschluss des IHK-Abschlussprojektes kann festgestellt werden, dass sämtliche Anforderungen, welche im Pflichtenheft festgelegt wurden, gemäß dem erweiterten Wasserfallmodell erfüllt wurden. Der erstellte Zeitplan im Abschnitt 2.1 (Projektphasen) konnte präzise eingehalten werden. Der Soll-Ist-Vergleich in Tabelle III zeigt, dass nur geringfügige Abweichungen von der geplanten Zeit aufgetreten sind. Die entstandenen Unterschiede konnten innerhalb des Projektes kompensiert werden, so dass das Projekt innerhalb des von der IHK vorgegebenen Zeitrahmens von 80 Stunden erfolgreich abgeschlossen werden konnte. Die Einplanung einer Zeitreserve konnte gut genutzt werden, da einige Phasen etwas mehr Zeit in Anspruch genommen haben, als erwartet.

Projektphase	Soll	Ist	Differenz
Analysephase	8h	8,5h	+ 0,5h
Entwurfsphase	18h	19h	+ 1h
Implementierungsphase	19h	20h	+ 1h
Testphase	10h	10,5h	+ 0,5h
Dokumentation	16h	16h	0h
Abschluss	6h	6h	0h
Zeitreserve	3h	0h	- 3h
Projektdauer	80h	80h	0h

Tabelle III – Soll-Ist-Vergleich

8.2 Lessons Learned

Im Rahmen dieses IHK-Abschlussprojektes wurden wertvolle Erfahrungen in Bezug auf die Planung und Umsetzung eines Softwareprojekts gesammelt. Es wurde deutlich, dass eine sorgfältige Planung und Analyse der Anforderungen entscheidend für den Erfolg des Projekts sind. Auch die Verwendung von klaren und einheitlichen Standards und Konventionen bei der Entwicklung sowie die Nutzung von geeigneten Werkzeugen und Frameworks kann die Effizienz und Qualität der Arbeit erheblich verbessern.

Auch die Beachtung von Fristen und die regelmäßige Überprüfung des Projektfortschritts waren essentiell, um das Projekt innerhalb des vorgegebenen Zeitrahmens erfolgreich abzuschließen. Im Laufe des Projekts wurden einige Herausforderungen identifiziert, wie z.B. größere Schwierigkeiten die Benutzeroberfläche so intuitiv wie möglich zu gestalten

ComBook*Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen*

oder die Darstellung der Büroräume. Diese Herausforderungen konnten durch eine flexible und proaktive Herangehensweise gemeistert werden.

Abschließend kann festgestellt werden, dass dieses Projekt wertvolle Einblicke und Erfahrungen in Bezug auf die Planung, Entwicklung und Umsetzung von Softwareprojekten geliefert hat, die in zukünftigen Projekten von großem Nutzen sein werden.

8.3 Ausblick

Für eine mögliche Erweiterung des Projektes kann in Betracht gezogen werden, das System um weitere Standorte zu erweitern. So könnte man das Zweitbüro in Berlin in Betracht ziehen. Hierbei müsste die Anwendung um zusätzliche Funktionalitäten erweitert werden, die die Verwaltung von mehreren Standorten und Räumen unterstützen.

Ein weiterer möglicher Ausblick wäre die Implementierung eines Parkplatz-Reservierungssystems. Hierbei könnte das bestehende System um zusätzliche Funktionalitäten erweitert werden, die es den Nutzern ermöglichen, Parkplätze in der Nähe des Arbeitsplatzes zu reservieren. Hierfür müssten zusätzliche Datenbanktabellen und Backend-Funktionen implementiert werden, um die Parkplatzreservierungen zu verwalten und zu verarbeiten.

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

9 Quellenverzeichnis

<https://www.modu-learn.de/verstehen/investition-finanzierung/amortisationsrechnung/>

(Stand 16.04.2023)

<http://stackoverflow.com/>

<https://de.reactjs.org/docs/getting-started.html>

10 Glossar

React: React ist eine JavaScript-Bibliothek, die auf einem Komponentenmodell basiert und zur Entwicklung von wiederverwendbaren Benutzeroberflächen für Webanwendungen verwendet wird.

C#: C# (C-Sharp) ist eine objektorientierte Programmiersprache, die von Microsoft entwickelt wurde und eine hohe Verbreitung in der Windows-Programmierung sowie bei der Entwicklung von Webanwendungen auf der .NET-Plattform hat.

MSSQL: MSSQL ist ein relationales Datenbankmanagementsystem, das von Microsoft entwickelt wurde und für die Verwaltung und Speicherung von strukturierten Daten in Unternehmen eingesetzt wird.

REST: REST (Representational State Transfer) ist ein Architekturstil für die Entwicklung von Webanwendungen, der auf HTTP-Protokollen basiert.

MVC-Pattern: Das MVC-Pattern (Model-View-Controller-Pattern) ist ein Entwurfsmuster zur Strukturierung von Softwareanwendungen.

JSON: JSON (JavaScript Object Notation) ist ein Datenaustauschformat, das auf JavaScript-Objekten basiert. Es ist einfach zu lesen und zu schreiben und wird häufig zur Übertragung von Daten zwischen Webanwendungen und APIs verwendet.

Entity-Relationship-Diagramm: Ein Entity-Relationship-Diagramm (ERD) ist ein visuelles Modell zur Darstellung von Datenbankstrukturen. Es zeigt die Beziehungen zwischen verschiedenen Entitäten und beschreibt ihre Attribute und Beziehungen zueinander.

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

11 Anhang

11.1 Detaillierte Zeitplanung

Analysephase	8 Std.
Ist-Analyse	2 Std.
Soll Konzept	2 Std.
Erstellung eines Pflichtenheft	2,5 Std.
Amortisationsrechnung	1,5 Std.
Entwurfsphase	18 Std.
Projektablaufplanung	2 Std.
Entwurf der Web-Oberfläche	6 Std.
Entwurf der Anwendungsarchitektur	7 Std.
Entwurf des Datenbank-Konzepts	3 Std.
Implementierungsphase	19 Std.
Implementierung der Weboberfläche	7,5 Std.
Implementierung der Anwendungsarchitektur	9 Std.
Implementierung des Datenbank-Konzepts	2,5 Std.
Testphase	10 Std.
Testen der Web-Oberfläche	2,5 Std.
Testen der Anwendungsarchitektur	3 Std.
Testen der Datenbank	1 Std.
Fehlerbehebung	2 Std.
Code-Review	1,5 Std.
Dokumentation	16 Std.
Projektdokumentation	10 Std.
Benutzerhandbuch	3 Std.
Entwicklerdokumentation	3 Std.
Abschluss	6 Std.
Abnahme durch den Auftraggeber	2 Std.
Durchführung eines Soll-Ist-Vergleichs	2,5 Std.
Deployment	1,5 Std.
Zeitreserve	3 Std.
Gesamt	80 Std.

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

11.2 Verwendete Ressourcen

Software

- Windows 10
- Visual Studio Professional 2022
- Webstorm 2022.3
- Tortoise-Git
- GitHub
- MS-SQL
- MySQL-Workbench 8.0
- IIS – Internet Information Services

Hardware

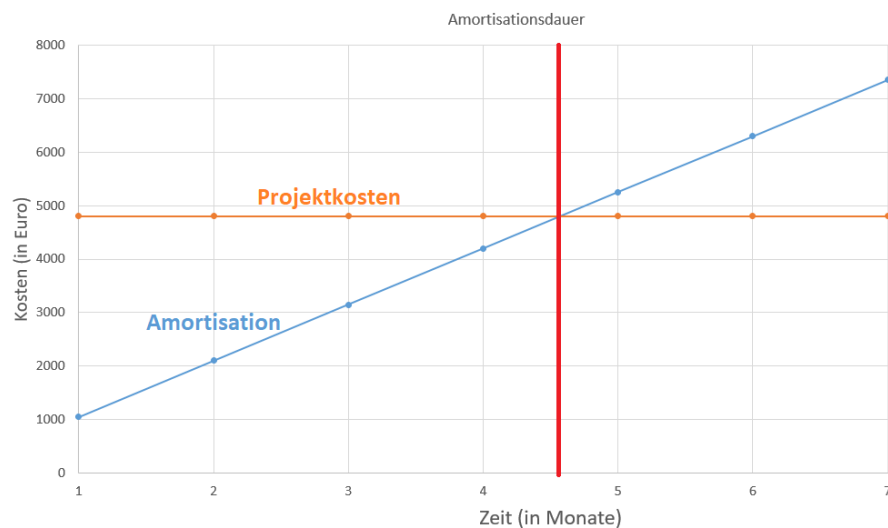
- Laptop mit Dockingstation
- 2x Bildschirm
- Tastatur
- Maus

Personal

- Auszubildender
- Mitarbeiter (Blackbox Test)
- Abteilungsleiter

11.3 Amortisation

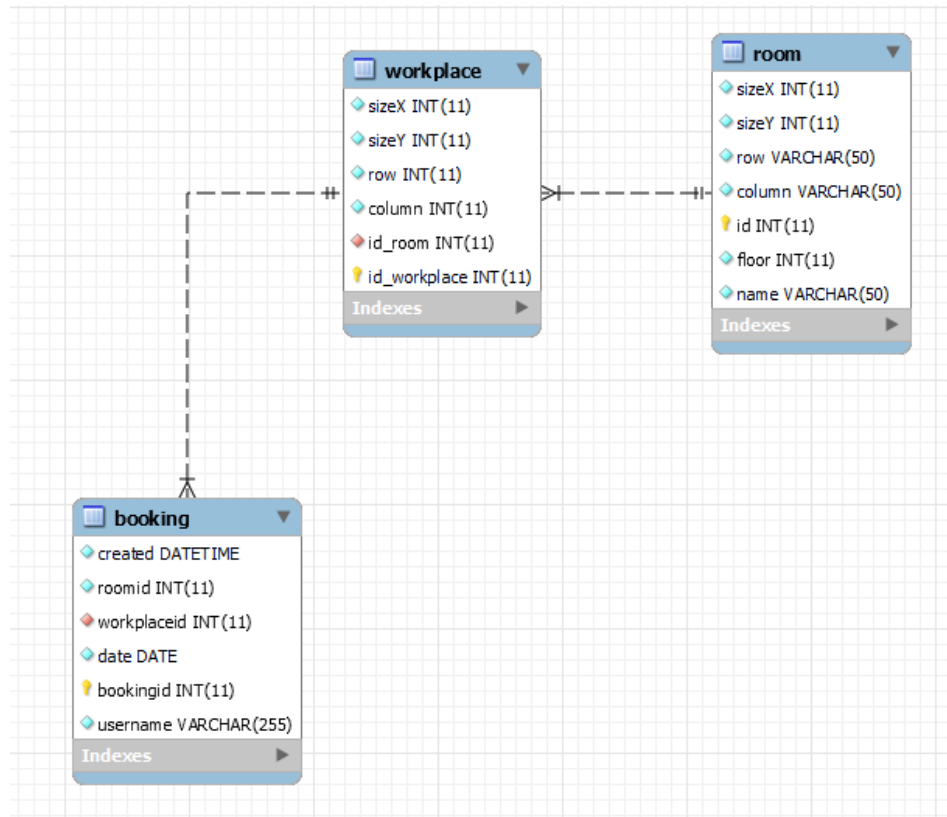
Der Schnittpunkt der Geraden gibt den Zeitpunkt der Amortisation an.



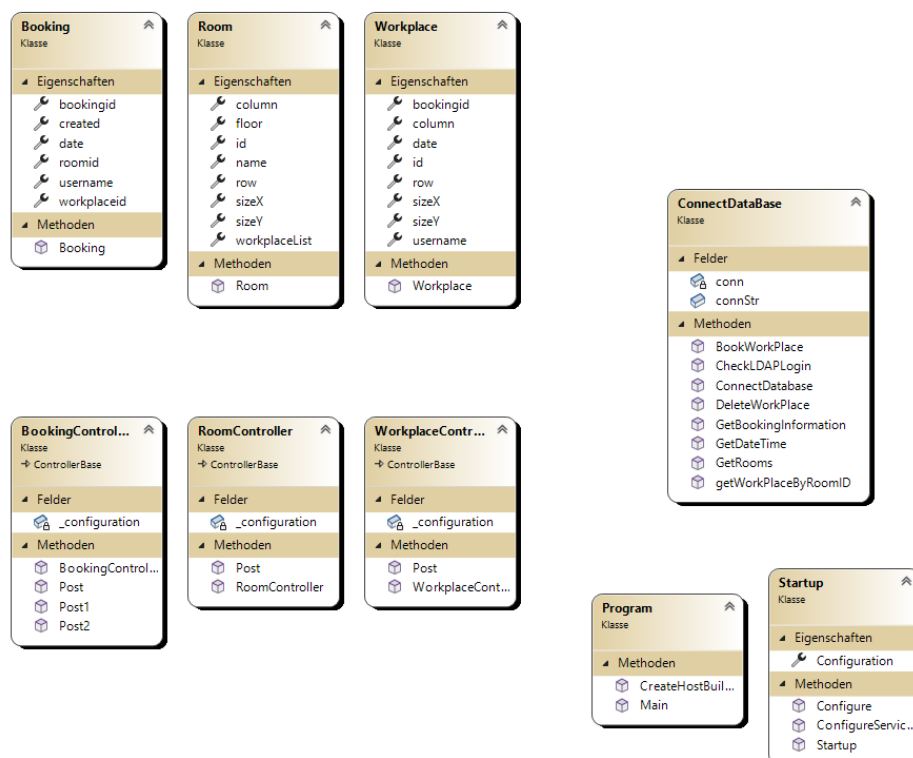
ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

11.4 Entity-Relationship-Diagramm



11.5 Klassendiagramm



ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

11.6 Benutzerhandbuch

Im Benutzerhandbuch wird Schritt für Schritt erklärt, wie man sich einloggt, einen Arbeitsplatz buchen kann und wie man diesen wieder stornieren kann. Hierzu folgt ein kurzer Ausschnitt aus dem Handbuch:



Benutzerhandbuch ComBook

<p>Das Login wird durch die Active Directory Zugangsdaten gesteuert:</p> 	<p>Das Buchungsdatum kann man mit Hilfe des Kalenders auswählen:</p> 	<p>Um sich zwischen den Etagen zu bewegen, kann man das Dropdown-Menü benutzen:</p> 	<p>Um einen Arbeitsplatz auszuwählen, kann man diesen mit einem Linksklick anklicken. Die Farbe des Arbeitsplatzes sollte sich nun zu grün ändern:</p> 
--	--	--	--

Um einen Sitzplatz zu buchen, kann man auf den Text/Knopf "SITZPLATZ BUCHEN" unter dem Kalender klicken. Der ausgewählte Arbeitsplatz wird nun Orange

Farbcodierung	Bedeutung
Weiß	Sitzplatz ist Frei
Grün	Sitzplatz wurde ausgewählt
Orange	Eigene Sitzplatzbuchung
Rot	Sitzplatzbuchung von einem anderen Mitarbeiter

Um nähere Infos zu den Buchungen zu erhalten oder seine eigene Buchung zu stornieren, kann man auf die Orangen oder Roten Arbeitsplätze klicken:

<p>Eigene Buchung (ORANGE)</p> 	<p>Buchung eines anderen Mitarbeiters (ROT)</p> 
---	--

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

11.7 Entwicklerdokumentation

Im folgenden Auszug der Entwicklerdokumentation wird ein Teil des Handbuchs zum C# Backend gezeigt:

Entwicklerhandbuch ComBook

Handbuch Backend C# ASP.NET Core 6 - Visual Studio 2022

Model:

Models sind ein nahezu 1 zu 1 Mapping von den jeweiligen Datenbankobjekten

Room

Konstruktor = int sizeX, int sizeY, string row, string column, int id, int floor, string name, List<Workplace> workplaces

Workplace

Konstruktor = int sizeX, int sizeY, int row, int column, int id, string date, long bookingid, string username

Booking

Konstruktor = DateTime created, int roomid, int workplaceid, string date, long bookingid, string username

Controller:

RoomController
WorkplaceController
BookingController

Controller stellen Schnittstellen zum Frontend da, welches via REST und JSON die Daten annimmt und verarbeitet.

Logik:

Methodenname	Beschreibung
ConnectDatabase()	Verbindung mit der Datenbank aufbauen
GetRooms(Room r)	Alle Räume einer bestimmten Etage aus der DB holen und zurückgeben
GetWorkPlaceByRoomID(int roomID, string date)	Alle Arbeitsplätze der bestimmten Räume holen und zurückgeben
BookWorkPlace(Booking booking)	Sitzplatz buchen
DeleteWorkPlace(Booking booking)	Sitzplatzbuchung stornieren
GetBookingInformation(Booking booking)	Informationen zu den jeweiligen Sitzplätzen holen und wiedergeben
GetDateTime()	Aktuelles Datum wiedergeben
CheckLDAPLogin(string username, string password)	Active Directory Zugang prüfen

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

11.8 Pflichtenheft

Im folgenden Auszug des Pflichtenhefts werden die Anforderungen, welche umgesetzt wurden, dokumentiert:

Umsetzung der Anforderungen

- Der Login, welcher über das Active Directory geregelt wird und nur Mitarbeiter des Unternehmens zulässt
- Graphische Oberfläche des Gebäudes mit den verschiedenen Etagen und Sitzplätzen
- Auswahl der Tage über einen Kalender
- Speicherung der Daten und Buchungen in der Datenbank
- Detaillierte Ansicht der Buchungen, wenn man auf diese klickt
- Möglichkeit, seine Buchung zu stornieren
- Benennung der einzelnen Büros in der Oberfläche

ComBook

Entwicklung einer Webanwendung zur Sitzplatzbuchung von Pool-Arbeitsplätzen

11.9 Auszug C# Quellcode

```
public static List<Room> GetRooms(Room r) {
    ConnectDatabase();

    List<Room> roomList = new List<Room>();

    conn.Open();
    string sql = "SELECT sizeX, sizeY, row, `column`, id, floor, name FROM room WHERE floor = ?floor";
    MySqlCommand cmd = new MySqlCommand(sql, conn);
    cmd.Parameters.AddWithValue("?", r.floor);

    var reader = cmd.ExecuteReader();

    if (reader.HasRows) {
        while (reader.Read()) {
            if (!reader.IsDBNull(0) && !reader.IsDBNull(1) && !reader.IsDBNull(2) &&
                !reader.IsDBNull(3) && !reader.IsDBNull(4) && !reader.IsDBNull(5) &&
                !reader.IsDBNull(6)) {
                int sizeX = reader.GetInt32(0);
                int sizeY = reader.GetInt32(1);
                string row = reader.GetString(2);
                string column = reader.GetString(3);
                int id = reader.GetInt32(4);
                int floor = reader.GetInt32(5);
                string name = reader.GetString(6);

                List<Workplace> workList = getWorkPlaceByRoomID(id,
                    r.workplaceList[0].date);
                Room room = new Room(sizeX, sizeY, row, column, id, floor, name, workList);
                roomList.Add(room);
            }
        }
        reader.Close();
        conn.Close();

        return roomList;
    }

    public static List<Workplace> getWorkPlaceByRoomID(int roomID, string date) {

        MySqlConnection conn = new MySqlConnection(connStr);

        conn.Open();

        List<Workplace> workList = new List<Workplace>();
        string sql = "SELECT sizeX, sizeY, `row`, `column`, id_workplace, COALESCE(bookingid, 0) AS bookingid, COALESCE(username, 0) AS username FROM workplace LEFT OUTER JOIN booking ON booking.workplaceid = workplace.id_workplace AND booking.date = ?date WHERE id_room = ?id";

        MySqlCommand cmd = new MySqlCommand(sql, conn);
        cmd.Parameters.AddWithValue("?", roomID);
        cmd.Parameters.AddWithValue("?", date);
        MySqlDataReader readerWorkplace = cmd.ExecuteReader();

        if (readerWorkplace.HasRows) {
            while (readerWorkplace.Read()) {
                if (!readerWorkplace.IsDBNull(0) && !readerWorkplace.IsDBNull(1) &&
                    !readerWorkplace.IsDBNull(2) && !readerWorkplace.IsDBNull(3) &&
                    !readerWorkplace.IsDBNull(4) && !readerWorkplace.IsDBNull(5) &&
                    !readerWorkplace.IsDBNull(6)) {
                    int sizeXworkplace = (int)readerWorkplace["sizeX"];
                    int sizeYworkplace = (int)readerWorkplace["sizeY"];
                    int rowworkplace = (int)readerWorkplace["row"];
                    int columnworkplace = (int)readerWorkplace["column"];
                    int idworkplace = (int)readerWorkplace["id_workplace"];
                    long bookingid = (long)readerWorkplace["bookingid"];
                    string username = readerWorkplace["username"].ToString();

                    Workplace workplace = new Workplace(sizeXworkplace, sizeYworkplace,
                        rowworkplace, columnworkplace, idworkplace, date, bookingid, username);
                    workList.Add(workplace);
                }
            }
            readerWorkplace.Close();
            conn.Close();
            return workList;
        }

        public static bool BookWorkPlace(Booking booking) {
            ConnectDatabase();

            conn.Open();
            string sql = "INSERT INTO `booking`(`created`, `roomid`, `workplaceid`, `date`, `username`) VALUES (?dateTime, (SELECT id_room FROM workplace WHERE id_workplace = ?workplaceId), ?workplaceId, ?bookingDate, ?username)";
```

[...]